WHAT IS CLAIMED IS:

1. An apparatus for re-ordering commands to access memory, the apparatus comprising:

an array controller having sequencers to service the commands;

a first penalty box to receive a first command of the commands, the first command being associated with a memory bank of the memory, and determine a penalty associated with the first command based upon a conflict between the first command and an access to the memory bank, the penalty to expire when the memory bank and a data bus associated with the memory bank are available to process the first command; and

a command queue to store the first command in a queue, update the penalty in response to a communication from the first penalty box and dispatch the first command to an available sequencer of the array controller after the penalty expires, to service the first command.

2. The apparatus of claim 1, further comprising a second penalty box to track a second penalty associated with the available sequencer in response to dispatching a second command to the available sequencer, wherein the second command has no conflict with accesses to the memory and the available sequencer is associated with an unexpired penalty.

3. The apparatus of claim 1, wherein the first penalty box comprises a fast path associator to associate a second command with a fast path buffer of the command queue, wherein the second command has no conflicts with accesses to the memory and a sequencer of the sequencers is available to service the second command.

4. The apparatus of claim 1, wherein the first penalty box comprises a penalty expiration associator to communicate the expiration of the first penalty to the command queue.

5. The apparatus of claim 1, wherein the command queue comprises

a load queue to store the first command when the first command is a load command;

a store queue to store the first command when the first command is store command; and

a *sequence tracker to associate the first command with the penalty.*

6. The apparatus of claim 5, wherein the sequence maintains an indication of the status of the penalty and identifies a sequencer of the sequencers that is associated with the penalty.

7. The apparatus of claim 1, wherein the command queue comprises a prioritization logic to select the first command from the command queue based upon a priority associated with the first command after the penalty expires, and to dispatch the first command to the available sequencer.

8. The apparatus of claim 1, wherein the command queue comprises an interface monitor to monitor commands dispatched from the command queue to the array controller, and to update associations between commands remaining in the command queue and the sequencers, based upon conflicts associated with the commands dispatched.

9. A method for re-ordering commands to access memory via an array controller having sequencers, the method comprising:

receiving a first command to access a memory bank of the memory;

determining a penalty associated with the first command based upon a conflict between the first command and an access to the memory bank, the penalty to expire when the memory bank and a data bus associated with the memory bank are available to process the first command;

queuing the first command; and

dispatching the first command to an available sequencer of the sequencers after the penalty expires, to service the first command.

10. The method of claim 9, further comprising:

monitoring for a processor intervention;

canceling the first command and initiating a store command in response to the processor intervention, wherein the processor intervention indicates that a modified content is associated with the first command and the first command is a load command; and

initiating a speculative load command associated with the first command in response to the processor intervention wherein the processor intervention indicates that a cache entry is invalid.

11. The method of claim 9, further comprising:

receiving a subsequent command having no conflict with accesses to the memory and

dispatching the subsequent command prior to dispatching the first command.

12. The method of claim 9, wherein determining a penalty comprises:

determining a latency associated with the first command, resulting from the access to the memory bank;

associating the latency with a sequencer of the sequencers that serviced the access;

associating the sequencer with the first command; and

decrementing the latency based upon passage of clock cycles in response to an indication that the access has been serviced, to determine when the penalty expires.

13. The method of claim 12, wherein queuing the first command comprises storing the status of the penalty associated with the first command and associating the status with a sequencer of the sequencers, wherein the sequencer serviced the access, to *determine when the status of the penalty changes.*

14. The method of claim 9, further comprising updating a penalty associated with the first command in response to a conflicting access to the memory bank created by servicing another command.

15. The method of claim 9, wherein queuing the first command comprises storing the first command in a load queue, wherein the first command comprises a load operation for the memory bank.

16. The method of claim 9, wherein queuing the first command comprises storing the first command in a store queue, wherein the first command comprises a store operation for the memory bank.

17. The method of claim 9, wherein dispatching the first command comprises selecting the first command after the penalty expires, based upon a priority associated with the first command.

18. A system for re-ordering commands, the system comprising:

a memory having more than one memory bank;

a memory controller coupled with the memory via a data bus, comprising

an array controller having sequencers to service the commands;

a first penalty box to receive a first command of the commands, the first command being associated with a memory bank of the memory, and to determine a penalty associated with the first command based upon a conflict between the first command and an access to the memory bank, the penalty to expire when the memory bank and the data bus are available to process the first command; and

a command queue to store the first command in a queue, update the penalty in response to a communication from the first penalty box and dispatch the first command to an available sequencer of the array controller after the penalty expires, to service the first command; and

a processor coupled with the memory controller to issue the commands to access the memory.

19. The system of claim 18, wherein the memory controller comprises a second penalty box to track a second penalty associated with the available sequencer in response to dispatching a second command to the available sequencer, wherein the second command has no conflict with accesses to the memory and the available sequencer is associated with an unexpired penalty.

20. The system of claim 18, wherein the memory controller comprises an intervention prioritizer to recognize an intervention initiated by the processor to indicate a modified content associated with a load command of a second processor, to cancel the load command, and to associate a priority with a store command for the modified content.

21. The system of claim 18, wherein the memory controller comprises an intervention prioritizer to recognize an intervention initiated by the processor and to initiate a speculative read for the processor in response to the intervention.